

**REMARKS**

Please reconsider the application in view of the above amendments and the following remarks. Applicant thanks the Examiner for carefully considering this application and for indicating that the drawings filed on November 14, 2003 are accepted.

**Disposition of Claims**

Claims 1-25 are pending in the present patent application. Claims 1, 11, 20, and 23 are independent. The remaining claims depend, either directly or indirectly, from claims 1, 11, 20, and 23.

**Claim Amendments**

Claims 11 and 20 have been amended to clarify the scope of the invention. Claims 12 and 16 are amended to conform to the amendments to claim 11. Support for the above amendment may be found, for example, in paragraphs [0029] and [0036] of the referenced application. No new matter has been added by any of the aforementioned amendments.

**Rejection(s) under 35 U.S.C. § 101**

Claims 11-22 stand rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. More specifically, the Examiner rejected claims 11-22 as reciting elements that fail to tangibly embody the system and, thus, are computer software per se. To the extent the rejection may apply to amended and original claims 11-22, the rejection is traversed.

Claim 11, as amended, recites, in part, “a processor configured to execute a tracing framework” and “a memory configured to store a property file.” The recited system comprising a processor and a memory is clearly not a program or software per se.

Claim 20, as amended, recites, in part, “[a] network system having a plurality of nodes, wherein each of the plurality of nodes comprises a processor and a memory.” The recited plurality of nodes, each node comprising a processor and a memory, are clearly not data structures per se.

For at least these reasons, claims 11 and 20 comply with the statutory subject matter requirement of 35 U.S.C. §101. Claims 12-19 and 21-22 depend either directly or indirectly from claims 11 and 20 and, thus, comply with the statutory subject matter requirement of 35 U.S.C. §101 for at least the same reasons as claims 11 and 20. Accordingly, withdrawal of this rejection is requested.

### **Rejection(s) under 35 U.S.C. § 112**

Claims 1-25 stand rejected under 35 U.S.C. § 112 as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. More specifically, the Examiner asserts that the specification fails to give an adequate description of “anonymous consumer state.” To the extent the rejection may apply to amended and original claims 1-25, the rejection is traversed.

The specification states that “[i]n one or more embodiments of the invention, the state of the tracing framework resulting from creating an anonymous consumer state from the object code loaded/written in the property file is the same as it would have been if a user-level program (*e.g.*, a

consumer) had defined a tracing operation and communicated it to the tracing framework during normal non-boot tracing.” *See* Referenced Application, [0030]. Further, the anonymous consumer state is associated with a consumer after the system has booted and user-level programs are able to run and direct kernel-level tracing frameworks. *See* Referenced Application, [0032]. Thus, the anonymous consumer state corresponds to a consumer state, which is present before the system is booted (or while the system is booting) and is equivalent to a consumer state present after the system is booted. In view of the above, the anonymous consumer state is adequately described in the specification.

For at least these reasons, claims 1-25 comply with the requirements of 35 U.S.C. §112. Accordingly, withdrawal of this rejection is requested.

### **Rejections under 35 U.S.C. § 102**

Claims 1-25 stand rejected under 35 U.S.C. § 102 as being anticipated by Tamches, “Fine-Grained Dynamic Instrumentation of Commodity Operating System Kernels”, University of Wisconsin, 2001 (Tamches). To the extent that this rejection may still apply to the amended and original claims, the rejection is respectfully traversed.

“A claim is anticipated only if *each and every element* as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.” *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631 (Fed. Cir. 1987) (emphasis added). Further, “[t]he identical invention must be shown in as complete detail as is contained in the claim.” *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236 (Fed. Cir. 1989). The Applicant respectfully asserts that

Tamches does not expressly or inherently describe each and every element of independent claims 1, 11, 20, and 23.

Independent claims 1 and 11 are directed to a method and apparatus for tracing an instrumented program. More specifically, claims 1 and 11 comprise, in part, loading object code defining enabling information into a *property file*. The Examiner has asserted that Tamches teaches loading object code into a property file. *See* Office Action mailed February 8, 2007, p. 3. The Applicant respectfully disagrees. Specifically, the Examiner has asserted that a “code patch heap” is equivalent to a “property file.” *See* Office Action mailed February 8, 2007, p. 3. By asserting that a code patch “heap” is equivalent to a property “file,” the Examiner is completely misconstruing the broadest ordinary meaning of the term property “file.”

Specifically, a “file” refers to “a collection of related data records” or “a complete collection of data (as text or a program) treated by a computer as a unit especially for purposes of input and output.” *See, e.g.,* Merriam-Webster's Collegiate® Dictionary: Eleventh Edition, 2003, as cited at <http://www.m-w.com>. Based on the above definition, a “file” refers to records stored as output resulting from the executing of a program or input to be used in the execution of a program.

In view of the above, a “heap” cannot be construed to be equivalent to a “file,” as a “heap” holds instrumentation code generated at run-time. *See* Tamches, pg. 49. Specifically, the instrumentation code is only a *representation* of the file in memory and is not the file itself. *See* Tamches, pg. 27-28 (describing how kernel modules are *relocatable files* that are loaded into memory). In view of this, it is clear that a code patch “heap” as described in Tamches is not equivalent to a property “file” as recited in independent claims 1 and 11.

Moreover, assuming *arguendo* that a heap is equivalent to a file, Tamches does not disclose creating an anonymous consumer state as recited in the independent claims 1 and 11. Specifically, claim 1 recites “processing the property file to enable the tracing framework, wherein enabling the tracing framework comprises creating an anonymous consumer state” and claim 11 recites “enabling information to create the anonymous consumer state.” In contrast, Tamches only discloses that kperfmon allows for profiling of a kernel to track the *performance* of a *kernel*. See Tamches, pg. 70. However, the profile information disclosed in Tamches does not disclose that the profile information is either anonymous or related to a consumer. In view of this, it is clear that profile information of Tamches is not equivalent an anonymous consumer state as recited in independent claims 1 and 11.

Finally, Tamches does not disclose “*rebooting* the system after loading the object code” as is disclosed in independent claim 1. In contrast, Tamches states that “KernInst is loaded and instruments the kernel entirely at run-time, *without* any need to recompile, *reboot*, or even pause the kernel.” Tamches, pg. 5. Thus, Tamches explicitly discloses that KernInst does not require the kernel to be rebooted in order to be instrumented and traced. In view of this, it is clear that dynamic loading which does not require a reboot of the kernel of Tamches is not equivalent to a method for tracing involving, in part, rebooting the system as recited in independent claim 1.

In view of the above, Tamches fails to disclose all the limitations of independent claim 1. Thus, claim 1 is patentable over Tamches. In addition, independent claim 23 includes at least the same patentable subject matter as claim 1 and, thus, is patentable over Tamches for at least the same reasons as claim 1. Dependent claims 2-10 and 24-25 are patentable for at least the same reasons as claim 1. Further, Tamches fails to disclose all the limitations of amended independent claim 11. In

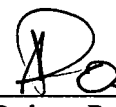
addition, amended independent claim 20 includes at least the same patentable subject matter as amended claim 11 and, thus, is patentable for at least the same reasons as claim 11. Dependent claims 12-19 and 21-22 are allowable for at least the same reasons as claim 11. Accordingly, withdrawal of this rejection is respectfully requested.

### Conclusion

Applicant believes this reply is fully responsive to all outstanding issues and places this application in condition for allowance. If this belief is incorrect, or other issues arise, the Examiner is encouraged to contact the undersigned or his associates at the telephone number listed below. Please apply any charges not covered, or any credits, to Deposit Account 50-0591 (Reference Number 03226/352001).

Dated: May 8, 2007

Respectfully submitted,

By  ROSSA <sup>\$ 60031</sup> ALY DOSSA  
Robert P. Lord  
Registration No.: 46,479  
OSHA · LIANG LLP  
1221 McKinney St., Suite 2800  
Houston, Texas 77010  
(713) 228-8600  
(713) 228-8778 (Fax)  
Attorney for Applicant